

智能温度传感器 Ds18B20 在多路测温中的应用

徐文进, 张阿卜

(厦门大学 自动化系 福建 厦门 361005)

摘要: 给出了总线上挂有多个 Ds18B20 的巡回测温的 C51 编程实现。通用性好, 特别适合用在大范围环境温度检测场合, 数据准确、可靠。最后给出笔者参与的一个应用项目。

关键词: 温度传感器; Ds18B20; 单总线; 多点测温

中图分类号: TP368.1

文献标识码: B

文章编号: 1004-373X (2004) 22-003-02

Application of Intelligent Thermometer Ds18B20 in Multiple Temperature Measurement

XU Wenjin, ZHANG Abu

(Department of Automation, Xiamen University, Xiamen, 361005, China)

Abstract: The program implemented with C51 programming language is presented in this paper. It can be applied to many fields especially in the large-scale environmental temperature measurement. It's accurate and reliable. Finally, a practical application developed by author is given.

Key words: thermometer; Ds18B20; single Bus; multiple temperature measurement

单总线芯片可以使得硬件结构非常简单, 特别适合应用在对于距离远、节点分布多的场合, 如粮仓温度检测, 大棚温室温度监控等。本文给出的设计通用性好, 可以方便地移植到其他应用场合。

1 Ds18B20 编程简介

每一片单总线芯片内部都有一个全球惟一的 64 b 编码, 在多路测温时就是通过匹配每个芯片的 ROM 编码 (ID), 来搜寻该路的温度。Ds18B20 有 9 个可擦写的内部寄存器, 称为便笺式 RAM。所有的串行通讯, 读写每一个 bit 位数据都必须严格遵守器件的时序逻辑来编程, 同时还必须遵守总线命令序列, 对单总线的 Ds18B20 芯片来说, 访问每个器件都要遵守下列命令序列:

首先是初始化; 其次执行 ROM 命令; 最后就是执行功能命令 (ROM 命令和功能命令后面以表格形式给出)。

如果出现序列混乱, 则单总线器件不会响应主机。当然, 搜索 ROM 命令和报警搜索命令, 在执行两者中任何一条命令之后, 要返回初始化。

基于单总线上的所有传输过程都是以初始化开始的, 初始化过程由主机发出的复位脉冲和从机响应的应答脉冲组成。应答脉冲使主机知道, 总线上有从机设备, 且准备就绪。

在主机检测到应答脉冲后, 就可以发出 ROM 命令。这些命令与各个从机设备的惟一 64 b ROM 代码相关。在

主机发出 ROM 命令, 以访问某个指定的 Ds18B20, 接着就可以发出 Ds18B20 支持的某个功能命令。这些命令允许主机写入或读出 Ds18B20 便笺式 RAM、启动温度转换。

2 软件实现

Ds18B20 的工作严格遵守单总线协议:

(1) 主机首先发一复位脉冲, 信号线上所有 Ds18B20 器件都被复位。

(2) 接着主机发送 ROM 命令, 程序开始读取单个在线的芯片 ROM 编码并保存在单片机数据存储器中, 然后卸下已读出的, 将未读出编码的 Ds18B20 逐个挂在总线上并读取其 ROM 编码, 直到把所有的需要用到的 Ds18B20 的 ROM 编码都离线读出, 最后用一个二维数组保存所有这些 ROM 编码, 数据保存在 X25043 中。

(3) 系统工作时, 把所有读取了编码的且需要用的 Ds18B20 挂在总线上。发温度转换命令, 再总线复位。

(4) 然后就可以从刚才的二维数组逐个匹配在线的温度传感器, 随后发温度读取命令就可以依次获得各路对应的温度值了。读取温度软件流程图如图 1 所示。

在主机初始化过程, 主机通过拉低单总线至少 480 μ s, 以产生复位脉冲。接着, 主机释放总线, 并进入接收模式。当总线被释放后, 上拉电阻将单总线拉高。在单总线器件检测到上升沿后, 延时 15~60 μ s, 接着通过拉低总线 60~240 μ s, 以产生应答脉冲。

```
//总线复位程序  
bit ds18b20_reset(void)
```

收稿日期: 2004-07-11

```

{ uchar data i;
  D0= 0; //发一个复位脉冲
  delayus(90); //延时 598  $\mu$ s
  D0= 1; //释放总线
  delayus(12); //等待 80  $\mu$ s 接受应答
  if(D0) return(0); //没有应答跳返回 0
  delayus(75); //延时 500  $\mu$ s 应答后等待器件释放总线
  return(1); //总线复位成功
}

```

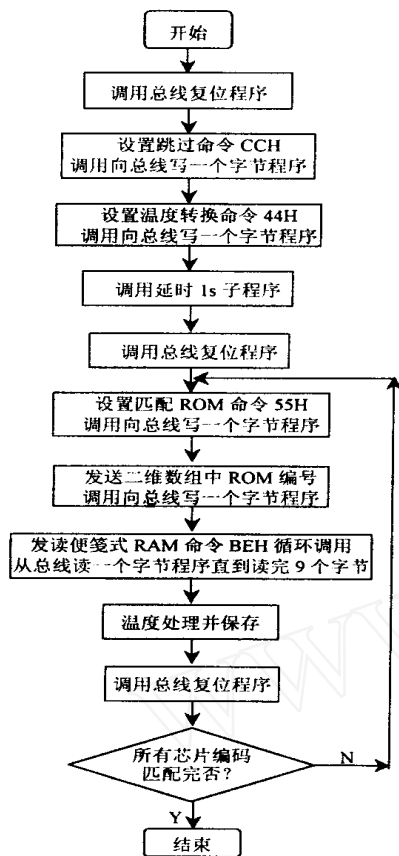


图 1 读取温度软件流程框图

写时序均起始于主机拉低总线,产生写 1 时序的方式:主机在拉低总线后,接着必须在 15 μ s 之内释放总线;而产生写 0 时序的方式:在主机拉低总线后,只需在整个时序期间保持低电平即可(至少 60 μ s)。

在写字节程序中的写一个 bit 位的时候,没有按照通常的分别写 0 时序和写 1 时序,而是把两者结合起来,当主机拉低总线后在 15 μ s 之内将要写的位 c 给 D0:如果 c 是高电平满足 15 μ s 内释放总线的要求,如果 c 是低电平,则 D0= c 这条语句仍然是把总线拉在低电平,最后都通过延时 58 μ s 完成一个写时序(写时序 0 或写时序 1)过程。

//主机写一个字节给 Ds18b20

```

void ds18b20_w_rite(uchar a)
{ uchar data i;
  bit c;
  for(i= 0; i< 8; i+ + )
  { c= (a>> i) & 0x01;
    D0= 0; //每次以主机拉低总线开始写时序
    D0= 0;

```

```

    delayus(0); //延时 13  $\mu$ s
    D0= c;
    delayus(7); //所有写序延时至少要 60  $\mu$ s
  }
  D0= 1; //写完一个字节释放总线
}

```

每个读时序都由主机发起,至少拉低总线 1 μ s,在主机发起读时序之后,单总线器件才开始在总线上发送 0 或 1。所有读时序至少需要 60 μ s。

//从 Ds18b20 读一个字节

```

uchar ds18b20_read(void)
{ uchar data i, a;
  for(i= 0; i< 8; i+ + )
  { D0= 0; //拉低总线
    D0= 0;
    //每次以主机拉低总线至少 1  $\mu$ s 表示器件可以开始发送 bit 位
    D0= 1; //释放总线主机准备接受 bit 位
    delayus(0); //等待 13  $\mu$ s
    a= (a>> 1);
    if (D0) a= a | 0x80; //读取总线数据位
    delayus(7); //所有读时序需要至少 60  $\mu$ s
  }
  D0= 1; //读完一个字节后释放总线
  return a;
}

```

以上 3 个子程序虽然是针对 Ds18b20 开发的,其实单总线芯片的时序都相似,只要根据其他类似芯片的时序特点对上述子程序稍做修改就可以同样使用。下面是多点测温主程序。输入参数 x 为测温点的号码。

//读取通道 x 的温度值

```

uint read_temp(uchar x)
{
  uchar i;
  uint idata temp;
  ds18b20_reset();
  ds18b20_w_rite(0xcc); //跳过 ROM 命令
  ds18b20_w_rite(0x44); //所有的传感器开始转换温度
  delayms(50);
  ds18b20_reset();
  ds18b20_w_rite(0x55); //匹配 ROM 命令
  for(i= 0; i< 8; i+ + )
  //rom id 二维数组存放每个芯片 ROM 编号
  ds18b20_w_rite(rom_id[x][i]);
  ds18b20_w_rite(0xBE);
  for(i= 0; i< 8; i+ + )
  //rom dat 存放读取到便笺式 RAM 的数据
  rom_dat[i]= ds18b20_read();
  ds18b20_reset();
  temp= rom_dat[1]; //温度高字节
  temp= (temp<< 8);
  temp+= rom_dat[0]; //温度低字节
  return (temp);
  //得到 int 型的温度,主程序中可以乘上基数 0.0625 就是准确的温度值
}

```

限于篇幅,这里没有列出读单个 Ds18b20 芯片 ROM 编号的程序。多路测温没有采用发 ROM 命令 0FH 搜索所

(下转第 7 页)

4 基于OV7648 图像采集系统

OV 7648 图像传感器得到众多后端DSP 商家的支持, 如 EPSON, CORELOGIC, ADI, ATI, AIC, SUNPLUS, WINBOND, ASANA, TI, EMBRAGE, APHAMOSIA, HITACHI, TOSHIBA 等。有了这些著名公司的支持, 不仅可以实现LCD 的驱动, JPEG 图像的压缩和解码, 甚至可以完成MPEG4 图像。

现以Omnivision 公司生产的后端DSP —— OV 528 为例说明OV 7648 的应用, 其原理框图如图3 所示。

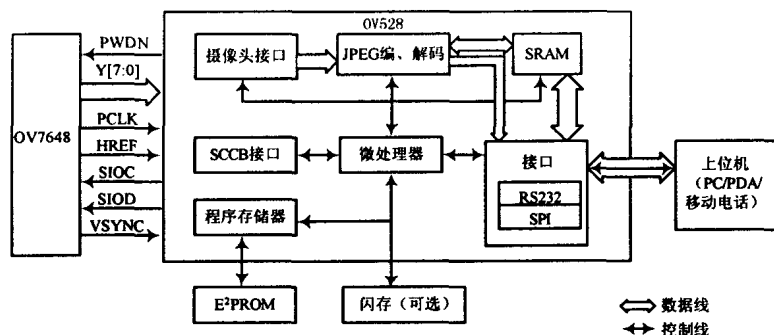


图3 基于OV 7648 图像采集系统原理框图

图3 中的PWN 是功耗模式选择, 低电平时为正常模式, 高电平时为低功耗模式。PCLK, HREF, VSYNC 分别为点频和行场同步输出信号线。OV 528 通过SCCB 接口对OV 7648 进行 Γ C 控制, 包括对比度、亮度、饱和度、白

平衡及自动曝光、同步信号位置及极性输出、帧速率和数据输出格式控制等。E²PROM 选用24C64。OV 528 是Omnivision 公司生产的一款单片低功耗、小体积、高性能的DSP, 主要用于PDA、移动电话的图像处理系统中, OV 528 通过摄像头接口从传感器OV 7648 中获取输入视频数据, 并能保证数据传输的实时性, 使用者在上位机(PC, PDA, 移动电话等) 上发出拍摄指令后, 系统捕获一帧静态图像送入OV 528 进行数据处理, 完成颜色校正、JPEG 压缩和解码等, 然后把数据送到SRAM 储存或直接通过接口传输到上位机, OV 528 支持的接口有两种: 一种是RS232, 波特率范围在9.6~920 kb/s 之间, 常用的波特率设置值是115.2 kb/s, 230.4 kb/s 和460.8 kb/s; 另一种是SPI, 可以获得更高的传输速率, 达到2~4 Mb/s。

该系统由于省略了A/D 等器件, 因而具有结构简单、体积小、功耗低等优点。同时, 由于OV 7648 内部嵌入曝光、增益、开窗等控制电路, 并且编程方便, 这就提高了采集系统在功能上的灵活性。本系统适合于监控、多媒体等应用范围。

参 考 文 献

- [1] Omnivision 公司产品技术手册. 2003.
- [2] [日] 谷口庆志. 数字图像处理 [M]. 朱虹, 廖学成, 乐静译. 北京: 科学出版社, 2002.

作者简介 周连恒 华南理工大学自动化学院硕士在读。主要研究方向为电力电子技术、嵌入式系统研究开发。

郝晓田 华南理工大学自动化学院副教授。主要研究方向为电力电子技术、运动控制系统、计算机控制、嵌入式系统等。

(上接第4页)

有在线温度传感器, 而是采用ROM 命令33H 先离线读出需要用到的每个温度传感器, 这样做不但简化了编程, 也不用进行在线的逐个搜索ROM 编码而浪费大量的时间等待搜索是否完成(平均每搜索75 个芯片ROM 编码需1 s)。通过离线一片一片地读出其中ROM 编码, 在存储这些编码的二维数组中也可以事先匹配好测温的对象。

3 应用实例

在混凝土大坝温度检测中, 需要实时而准确地显示温度变化情况, 将温度传感器埋需要温度监控的位置, 每个测温模块在需要重点监控温度的地方都放置了测温芯片, 而测温模块分布很广, 所以单片机采用RS485 总线形式与上位机通讯, 该总线为主从式, 通讯接口用RS485 芯片MAX1487, 经验证, 当波特率选为9 600 kb/s 时, 通讯距离可在2 km 左右范围正常工作, 上位机发命令轮巡读

取下位机端温度, 下位机之间不传递数据, 下位机收到上位机发来的命令, 启动测温点温度转换, 转换完后送上位机, 并在LED 上显示, 整个工作由上位机程序和下位机串口中断协调完成, 该系统测温值稳定、精确。

4 结 语

在大范围多点测温系统中, 这种分布式温度监控系统具有广阔的应用前景。

参 考 文 献

- [1] 沙占友. 智能化集成温度传感器原理与应用 [M]. 北京: 机械工业出版社, 2002.
- [2] Dallas Corp. Ds18B20 Programmable Resolution 1-wire Digital Thermometer.

作者简介 徐文进 男, 1981 年出生, 厦门大学自动化系硕士研究生。主要研究方向为计算机控制。

张阿卜 男, 1946 年出生, 教授, 厦门大学自动化系。